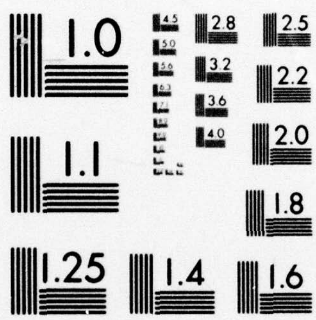
AD-A050 165 FLORIDA UNIV GAINESVILLE DEPT OF INDUSTRIAL AND SYS--ETC F/G 12/2
FINDING EFFICIENT SOLUTIONS FOR RECTILINEAR DISTANCE LOCATION P--ETC(U)
APR 77 L G CHALMET, R L FRANCIS DAHC04-75-G-0150
UNCLASSIFIED RR-77-3 ARO-12640.21-M NL

| OF |

AD
A050165

END
DATE
FILMED
3-78
DDC


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
6.3

10

# FINDING EFFICIENT SOLUTIONS FOR RECTILINEAR DISTANCE LOCATION PROBLEMS EFFICIENTLY

Research Report No. 77-3

by

Luc G. Chalmet*
Richard L. Francis**

April, 1977

Department of Industrial and Systems Engineering**
University of Florida
Gainesville, Florida 32611

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

This research was sponsored in part by *The Interuniversity College for Ph.D. Studies in Management Sciences (C.I.M), Brussels, Belgium, and by **the Army Research Office, Triangle Park, NC, under contract number DAHC04-75-G-0150.

* Catholic University of Louvain, Heverlee, Belgium.

THE FINDINGS OF THIS REPORT ARE NOT TO BE CONSTRUED AS AN OFFICIAL DEPARTMENT OF THE ARMY POSITION, UNLESS SO DESIGNATED BY OTHER AUTHORIZED DOCUMENTS.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

# REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

**1. REPORT NUMBER** 77-3 (19) 18 ARO 12640.21-M

**2. GOVT ACCESSION NO.**

**3. RECIPIENT'S CATALOG NUMBER** 9 Research rept.

**4. TITLE (and Subtitle)** 6 Finding Efficient Solutions for Rectilinear Distance Location Problems Efficiently,

**5. TYPE OF REPORT & PERIOD COVERED** Technical

**6. PERFORMING ORG. REPORT NUMBER** 14 RR-77-3

**7. AUTHOR(s)** 10 Luc G. Chalmet, Richard L. Francis

**8. CONTRACT OR GRANT NUMBER(s)** 15 DAHC04-75-G-0150

**9. PERFORMING ORGANIZATION NAME AND ADDRESS**
Industrial & Systems Engineering
University of Florida
Gainesville, Florida 32611

**10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS** 16 20061102A14D Rsch in & Appl of Applied Math.

**11. CONTROLLING OFFICE NAME AND ADDRESS**
U.S. Army Research Office
P.O. Box 12211
Triangle Park, NC 27709

**12. REPORT DATE** 11 Apr[redacted]77

**13. NUMBER OF PAGES** 38 12 39p.

**14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)**

**15. SECURITY CLASS. (of this report)**

**15a. DECLASSIFICATION/DOWNGRADING SCHEDULE**

**16. DISTRIBUTION STATEMENT (of this Report)**

Approved for public release; distribution unlimited

**17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)**

N/A

**18. SUPPLEMENTARY NOTES**

**19. KEY WORDS (Continue on reverse side if necessary and identify by block number)**

facility location — rectilinear distance
efficient locations — n-log-n algorithm

**20. ABSTRACT (Continue on reverse side if necessary and identify by block number)**


Given n planar existing facility locations, a planar new facility location X is called efficient if there is no other location Y at least as close to every existing facility as X, and strictly closer than X to at least one existing facility. ~~We present~~ an algorithm is presented which is either of order n(log n) or order n (depending upon how the problem is defined) that constructs all efficient locations, and establish that no alternative algorithm can be of a lower order. With the exception of two computational complexity results, ~~our~~ this work


next page

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

404 399

Jck

20.

is entirely self-contained, and relies almost entirely upon simple geometrical analyses.

TABLE OF CONTENTS

## ABSTRACT

Given n planar existing facility locations, a planar new facility location X is called efficient if there is no other location Y at least as close to every existing facility as X, and strictly closer than X to at least one existing facility. We present an algorithm which is either of order n(log n) or order n (depending upon how the problem is defined) that constructs all efficient locations, and establish that no alternative algorithm can be of a lower order. With the exception of two computational complexity results, our work is entirely self-contained, and relies almost entirely upon simple geometrical analyses.

## INTRODUCTION

Suppose a number of existing facilities are given, having planar locations $P_1, \ldots, P_n$. A new facility is to be located in the plane at some point X to be determined. With $P_i = (a_i, b_i)$, $X = (x, y)$ we denote the rectilinear distance between X and $P_i$ by $r(X, P_i)$ where, by definition.

$$r(X, P_i) = |x - a_i| + |y - b_i|.$$

Given any point Y in the plane for which

$$r(Y, P_i) \leq r(X, P_i), \ 1 \leq i \leq n,$$

we say that Y <u>dominates</u> X if X and Y satisfy the n inequalities, with at least one inequality holding strictly. In other words, Y dominates X if Y is at least as close to every existing facility location as X is, and closer than X to at least one existing facility location. If no point in the plane dominates a point X*, we say that X* is <u>efficient</u>. We denote the set of all efficient points by S*, and call S* the <u>efficient set</u>. Note that each existing facility location $P_j$ is in S*, for to have a point Y dominate $P_j$ would imply $r(Y, P_j) \leq r(P_j, P_j) = 0$, in turn implying $Y = P_j$.

Recently Wendell, Hurter, and Lowe [ 8 ] have introduced and studied the problem of finding $S^*$, and discuss some application contexts, with emphasis upon multiple objective problems. We remark that $S^*$ may also be of value in carrying out sensitivity analyses for single objective location problems, since such problems typically have the property that their optimal solutions are efficient. Also it appears that $S^*$ may be of value in the study of some internal warehouse location problems, in which case each existing facility would be a warehouse dock, and the new facility location would be the location of an item in the warehouse.

Wendell, Hurter, and Lowe develop a number of properties of $S^*$, and present two different algorithms for constructing $S^*$. Their work relies upon a good deal of relatively deep convexity analysis [2], [7]. We establish in this paper that $S^*$ can be characterized in an entirely self-contained, simple, and intuitively appealing manner, using only geometry. We consider our work both complements and supplements the work of Wendell, Hurter, and Lowe. In particular, we point out that the arrow algorithm we present for constructing $S^*$ is closely related to, and motivated by, the second of the two algorithms in [8].

The primary value of the arrow algorithm in our approach is as a tool to facilitate proofs. A second algorithm we present, the row algorithm, is more efficient. In fact, the row algorithm is the most efficient possible, in the sense that there exists no algorithm to construct $S^*$ which has a smaller order of computational effort, $n(\log n)$, than the row algorithm. For example, efficient implementations of the first and second algorithms in [8] result respectively in computational orders of $n^2$ and $n^3$; either order is greater than that of the row algorithm. For a discussion of a number of other algorithms of order $n(\log n)$ for solving geometric problems, see Shamos [5], [6].

Subsequently, following [8], we define the Line Construction Procedure: roughly speaking the procedure consists of plotting the points $P_1, \ldots, P_n$ and constructing both a vertical and horizontal line through each point. If the algebraic equivalent of the line construction procedure is considered to be part of the problem formulation, rather than part of the row algorithm, then the row algorithm is of order $n$, and no algorithm to construct $S^*$ can be of an order smaller than n. Even when the line construction procedure is considered to be part of the row algorithm, the algorithm performs as if it is of order n until n becomes "large."

In the next section, after presenting and illustrating a number of definitions, we present and illustrate the "arrow" algorithm which constructs the efficient set. We then give a characterization of the efficient set. The following section contains the row algorithm. The last section of the paper consists of the analysis needed to justify the algorithms and the characterization of $S^*$.

## CHARACTERIZING THE EFFICIENT SET

Figure 1 illustrates a basic notion, a diamond with center $P_i$ and radius $e_i$, denoted by $D(P_i, e_i)$. $D(P_i, e_i)$ consists of all points in the plane whose rectilinear distance from $P_i$ is no greater than $e_i$. The boundary of $D(P_i, e_i)$ consists of all points in the plane whose distance is equal to $e_i$, and so of course has the property that any two points on the boundary are the same rectilinear distance ($e_i$) from $P_i$. In other words, the boundary is a contour line of the rectilinear distance from $P_i$ of value $e_i$.

We call line segments parallel to the line $y = x$ SW - NE line segments, and call line segments parallel to the line $y = -x$ SE - NW line segments. Note that two edges of any $D(P_i, e_i)$ are SE - NW line segments, while the other two edges are SW - NE line segments.

Line Construction Procedure. Through each point $P_i$ construct a horizontal line and a vertical line. The horizontal (vertical), line should extend at least as far right and as far left (as far up and as far down) as every $P_i$. Subsequently whenever we refer to a line we mean a constructed line unless we specify otherwise. Figure 2a illustrates the construction procedure.

Noncollinearity Assumption. We assume that not all the $P_i$ lie on a single vertical line, or on a single horizontal line, as in this case $S^*$ is just the line segment joining the two $P_i$ which are farthest apart, so that constructing $S^*$ is a trivial problem.

Definitions. Figure 3 illustrates a number of the definitions to follow. For any vertical line we define the union of the line with the set of points to the right (left) of the line to be the set of points which are <u>east</u> (<u>west</u>) of the line. (Note that this definition permits a point on a vertical line to be both east and west of the line.) Similarly we define the set of points <u>north</u>, and the set of points <u>south</u>, of each horizontal line.

Given any two distinct adjacent horizontal lines H and H', with H north of H', and any two distinct adjacent vertical lines V and V', with V east of V', we call the set of points lying west of V, east of V', south of H, and north of H', a <u>box</u>, and denote the box by B. We call the collection of all boxes between any two adjacent vertical (horizontal) lines a <u>column</u> (<u>row</u>). Each of the four intersections of the box with a line we call an <u>edge</u> of B. We say two boxes are <u>adjacent</u> if their intersection is an edge of each box. The collection of all points lying south of H' and east of V we call the <u>SE direction</u> of B (abbreviated SE(B)) similarly we define <u>SW</u>, <u>NW</u>, and <u>NE</u> <u>directions</u> of B, and use the abbreviations SW(B), NW(B), and NE(B) respectively. We call the abbreviations SE, SW, NW, and NE the <u>box</u> <u>direction</u> <u>labels</u>. We say that a direction of B is <u>unoccupied</u> (<u>occupied</u>) if there is no (at least one) $P_i$ in the direction. We denote the union of all the boxes by $\beta$.

<u>Arrow Drawing Procedure</u>. For each box B we draw an arrow pointing from SE(B) to NW(B) whenever SE(B) is unoccupied, and call the arrow a SE <u>arrow</u>. We say the arrow <u>points</u> <u>away</u> <u>from</u> the south and east edges of B, and <u>points</u> <u>towards</u> the west and north edges of B. Likewise we construct and define SW, NW, and NE arrows whenever SW(B), NW(B), and NE(B) respectively is unoccupied. We call the abbreviations SE, SW, NW, NE the <u>arrow</u> <u>direction</u> <u>labels</u>. Figure 2b illustrates the arrow drawing procedure.

The following observation provides the motivation for the arrow drawing procedure.

<u>Observation 1</u>. Suppose we are given any box B with arrow $\bar{\alpha}$ in B. For any point X in B, move the point X in B in the direction towards which $\bar{\alpha}$ points along a line segment parallel to $\bar{\alpha}$, until the point intersects a box edge at a point X'. X and X' are such that $r(X', P_i) \leq r(X, P_i)$, $1 \leq i \leq n$. Further, if $X' \neq X$ and the direction towards which $\bar{\alpha}$ points is occupied, then X' dominates X.

To establish this observation (see Figure 4 ), suppose without loss of generality, that $\bar{\alpha}$ is a SE arrow and $X' \neq X$. We have $r(X', P_i) = r(X, P_i)$ for $P_i$ in NE(B) $\cup$ SW(B), while $r(X', P_i) < r(X, P_i)$ if $P_i$ is in NW(B). Since SE(B) is unoccupied, X' thus dominates X whenever NW(B) is occupied.

<u>Observation 2</u>. Each box has exactly 0, 1, or 2 arrows. Whenever a box has 2 arrows the arrows are parallel and point in opposite directions.

To establish this observation we note that if a box has perpendicular arrows then there are no $P_i$ in some half-plane defined by a line passing through some edge of the box, which is impossible. Hence a box has at most 2 arrows, and if it has 2 arrows then the arrows are parallel.

<u>Definitions</u>. We call a box with 0, 1, or 2 arrows a <u>null-box</u>, <u>1-box</u>, and <u>2-box</u> respectively. For any 1-box B we call the two edges towards which the arrow in B points the <u>leading edges</u> of B (see Figure 4).

By virtue of the definitions and Observation 2, we have

<u>Observation 3</u>. Each box in B is either a null-box, 1-box, or 2-box. A null-box has no unoccupied directions. A 1-box has exactly one

unoccupied direction, which has the same direction label as the box arrow. A 2-box has exactly two unoccupied directions, which differ by 180 degrees: the labels of the unoccupied directions are identical to the labels of the two arrows in the box.

Figure 5 illustrates Observation 3.

Observations 1 and 3 give

Observation 4. If B is a 1-box, and X is any point in B such that X is not on a leading edge of B, then there is a point X' on a leading edge of B such that X' dominates X.

It is also convenient to state

Observation 5. Any point X which is not in $\beta$ is dominated by a point X' which lies on the boundary of $\beta$, and is the closest point in $\beta$ to X.

We now state the

Arrow Algorithm. To determine the set S* of all efficient locations, carry out the line construction and arrow drawing procedures, and classify each box as a null-box, 1-box, or 2-box.

If there are no 1-boxes, take $S^* = \beta$; otherwise choose a 1-box B not yet chosen and delete from $\beta$ all points in B except those on the leading edges of B: repeat this deletion procedure for every 1-box. Denote by $\bar{\beta}$ the subset of $\beta$ remaining after the completion of the deletion procedure. Take $S^* = \bar{\beta}$.

We remark that if E is a common edge of two 1-boxes, B and B', if E is a leading edge of B, and not a leading edge of B', then E (except for one endpoint) will be deleted from $\beta$ once B' is chosen. Figures 2a, 2b, and 2c illustrates the algorithm: a null-box is identified by a dot in the box.

Due to Observations 4 and 5, we have

Observation 6. Any point not in $\bar{\beta}$ is dominated by a point in $\bar{\beta}$.

With two additional definitions we can characterize $S^*$.

Definition. An edge E of any box B is called a connecting edge if

(a) every arrow in B points towards E and E is contained in the boundary of $\beta$, or

(b) there is also a box, say B', such that E is the common edge of B and B', and every arrow in each box points towards E.

Connecting edges are illustrated in Figure 6.

Definition. Denote by $\beta^*$ the union of the following:

(a) all null-boxes

(b) all 2-boxes

(c) all connecting edges.

Subsequently we establish that $S^*$, the set of efficient points, $\tilde{\beta}$, the set of points left by the algorithm, and $\beta^*$ are all identical. Thus the arrow algorithm deletes all points in $\beta$ which are not points in $\beta^*$. Figure 2c illustrates $\beta^*$.

As a final comment, we note that the arrow algorithm as stated, and each of the algorithms in [8], is "memoryless" to some extent, in the sense that each algorithm can make more use than it does of information obtained during the process of determining the efficient set. In the following section we present the row algorithm, which exploits such information efficiently.

## CONSTRUCTING THE EFFICIENT SET: THE ROW ALGORITHM

Some notation is convenient. Denote the horizontal lines by $H_1$, $H_2$, ..., $H_{p+1}$ from north to south, and the rows by $R_1$, ..., $R_p$ from north to south. For $1 \leq i \leq p+1$, denote by $W_i(E_i)$ the x coordinate of the westmost (eastmost) existing facility location on $H_i$. For $R_i$, $1 \leq i \leq p$, define $NW_i(SW_i)$ to be the x coordinate of the westmost existing facility location which is north (south) of $R_i$. Define $NE_i(SE_i)$ to be the x coordinate of the eastmost existing facility location which is north (south) of $R_i$. (See Figure 11.)

As an immediate consequence of the definitions we have

Observation 7 Let B be a box in $R_i$, $1 \leq i \leq p$.

(a) NW(B) [SW(B)] is unoccupied if and only if B is west of the vertical line $x = NW_i (x = SW_i)$.

(b) NE(B) [SE(B)] is unoccupied if and only if B is east of the vertical line $x = NE_i (x = SE_i)$.

By virtue of the above observation we can readily classify the boxes in each row $R_i$. The following observation facilitates the computations of $NW_i$, $SW_i$, $NE_i$, and $SE_i$.

Observation 8 The following recursive relationships are true:

$$NW_1 = W_1 \qquad NE_1 = E_1$$

$$NW_i = \min(W_i, NW_{i-1}) \qquad NE_i = \max(E_i, NE_{i-1}) \quad , 2 \leq i \leq p$$

$$SW_i = \min(W_{i+1}, SW_{i+1}) \qquad SE_i = \max(E_{i+1}, SE_{i+1}) \quad , 1 \leq i \leq p-1$$

$$SW_p = W_{p+1} \qquad SE_p = E_{p+1}.$$

The above recursions are easily established. For example, certainly $NW_1 = W_1$. Also, the only existing facility locations north of $R_i$ lie on $H_1, H_2, \ldots, H_i$, and so

$$NW_i = \min(W_1, W_2, \ldots, W_i).$$

Likewise

$$NW_{i-1} = \min(W_1, W_2, \ldots, W_{i-1})$$

and so certainly

$$NW_i = \min(W_i, NW_{i-1}).$$

An equivalent geometric means of computing the recursions, which both provides insight and is easy to carry out manually, may be described as follows.

4-Color Procedure. Associate the colors Blue, Green, Red, and Yellow with the directions NW, NE, SW, and SE respectively.

Repeat the follwoing North to South Step for lines $H_1, \ldots, H_{p+1}$ consecutively. Beginning at the west (east) boundary of $\beta$, draw a blue (green) line over $H_i$ from west to east (east to west) terminating the line at $x = NW_i$ ($x = NE_i$), which is the point at which one of the following two events first occurs:

(a) The line initially intersects an existing facility location on $H_i$,

(b) The line attains the same length as the blue (green) line on $H_{i-1}$.

Repeat the following South to North Step for lines $H_{p+1}, \ldots, H_1$ consecutively. Beginning at the west (east) boundary of $\beta$, draw a red (yellow) line over $H_i$ from west to east (east to west), terminating the line at $x = SW_i$ ($x = SE_i$), which is the point at which one of the following two events first occurs:

(a) The line initially intersects an existing facility location on $H_i$,

(b) The line attains the same length as the red (yellow) line on $H_{i+1}$.

Note, in the north to south step that when an existing facility lying on the west (east) boundary of $\beta$ and $H_i$ is first encountered, every blue (green) line drawn subsequently is a degenerate line of zero length. In the south to north step when an existing facility lying on the west (east) boundary of $\beta$ and $H_i$ is first encountered, every subsequent red (yellow) line drawn is a degenerate line of zero length. Figure 11 illustrates the use of the 4-color procedure.

<u>Classifying the Boxes</u>.

Once the 4-color procedure has been completed, if we suppose the initial horizontal lines to be uncolored, it is easily verified that the north edge of each box colored in the north to south step is exactly one of the colors blue or green, while the south edge of each box colored in the south to north step is exactly one of the colors red or yellow. A box is a NW 1-box (NE 1-box) if and only if its north edge is blue (green) and its south edge is either uncolored or blue (uncolored or green). A box is a SW 1-box (SE 1-box) if and only if its south edge is red (yellow) and its north edge is either uncolored or red (uncolored or yellow). A box is a NW-SE (NE-SW) 2-box if and only if its north and south edges are blue and yellow (green and red) respectively. A box is a null-box if and only if both its north and south edges are uncolored. The color combinations listed in this paragraph for the three types of boxes are the only ones possible.

<u>Definitions</u>.

(i) We observe that $NW_i(SW_i)$ is the projection onto the x axis of the east tip of the blue (red) line on $H_i$ $(H_{i+1})$, while $NE_i$ $(SE_i)$ is the projection onto the x axis of the west tip of the green (yellow) line on $H_i$ $(H_{i+1})$. We refer to $NW_i$, $SW_i$, $NE_i$, and $SE_i$ as the blue, red, green, and yellow projections for row i. For each row i, we define

$$W_i^* = \max(NW_i, SW_i)$$

$$E_i^* = \min(NE_i, SE_i)$$

and call $W_i^*$ and $E_i^*$ the <u>west</u> and <u>east</u> <u>projections</u> for row i. See Figure 11 for an example.

(ii) Given a box B and distinct vertical lines $x = V'$ and $x = V$ with $V'$ west of V, we write $V' < B < V$ when we mean $V'$ is west of B and B is west of V. Note that for a given row, there is at least one box B in the row such that $V' < B < V$ if and only if $V' < V$. Further, for a given box B and vertical line $V''$, either $B < V''$ or $V'' < B$.

(iii) Comparisons of $W_i^*$ and $E_i^*$ for $R_i$ are crucial to the row algorithm we shall develop for finding S*. Define $R_i$ to be in <u>Condition 0</u> (C-0), <u>Condition 1</u> (C-1), or <u>Condition 2</u> (C-2) when $W_i^* < E_i^*$, $W_i^* = E_i^*$, and $W_i^* > E_i^*$ respectively. Note the three conditions are mutually exclusive and exhaustive.

(iv) For each line $H_i$, $1 \leq i \leq p + 1$, define

$$u_i = \max(NW_i, SW_{i-1}) \qquad v_i = \min(NE_i, SE_{i-1})$$

where, by convention, $SW_o = -\infty$, $SE_o = \infty$.

<u>Interpreting the Conditions</u>.

<u>(C-0)</u> Since a box in $R_i$ is a null-box if and only if its north and south edges are uncolored, the following conditions are all equivalent for at least one box B in $R_i$ to be a null-box: the projection of B lies between the west and east projections for $R_i$; $W_i^* < B < E_i^*$; $W_i^* < E_i^*$; $R_i$ is in C-0. When $W_i^* < E_i^*$ the null-boxes in $R_i$ are those boxes B for which $W_i^* < B < E_i^*$.

<u>(C-2)</u>. Since a box in $R_i$ is a 2-box if and only if its north and south edges are either blue and yellow or green and red respectively, there is a least one 2-box in $R_i$ if and only if the west and east projections for $R_i$ overlap, that is, $W_i^* > E_i^*$, that is, $R_i$ is in C-2. Subsequently we establish there is exactly one 2-box B in $R_i$ if and only if $E_i^* < B < W_i^*$.

<u>(C-1)</u>. Denote by V the vertical line identical to the line $x = W_i^* = E_i^*$. Denote by E that part of V lying south of $H_i$ and north of $H_{i+1}$. Since $W_i^* = E_i^*$ the west and east projections meet, so each of the boxes of which E is a vertical edge has at least one colored edge. If E is an east vertical edge of B, at least

one horizontal edge of B is blue or red, while if E is a west vertical edge of B', at least one horizontal edge of B' is green or yellow. Thus an arrow of each box in $R_i$ of which E is a vertical edge points towards E, so that E is a vertical connecting edge. Conversely, we establish subsequently that if E is a vertical connecting edge in $R_i$ lying on the line x = V, then $W_i^* = V = E_i^*$.

Horizontal Connecting Edges.

We remark it can be shown that a horizontal box edge is a horizontal connecting edge if and only if it is uncolored, and not an edge of any null-box. In order to identify the uncolored portion (if any) of each line $H_i$, in the north to south (south to north) step of the 4-color procedure we note that the blue (red) coloring of $H_i$ from west to east stops at the point where $H_i$ intersects the line $x = NW_i$ ($x = SW_{i-1}$). Thus the blue and/or red coloring of $H_i$ from west to east stops at the point where $H_i$ intersects the line $x = u_i$. Similarly, the green and/or yellow coloring of $H_i$ from east to west stops at the point where $H_i$ intersects the line $x = v_i$. Thus when $u_i = v_i$ all of $H_i$ is colored, while when $u_i < v_i$ the x projection of the uncolored portion of $H_i$ is the interval $[u_i, v_i]$.

Motivation for the Row Algorithm.

It is a readily established fact that if B is a NW 1-box, and B' is any box which is not east of B, and not south of B, then B' is also a NW 1-box. Completely analogous statements are true for NE, SE, and SW 1-boxes. Thus, roughly speaking, NW, NE, SE, and SW 1-boxes should be in the NW, NE, SE, and SW portions respectively of $\beta$. Effectively, the row algorithm implicity deletes such 1-boxes from $\beta$, and then what is left becomes $\beta*$.

We now state formally the algorithm based on the 4 color procedure.

Row Algorithm.

(1) Rank the existing facilities by their y coordinates to determine the lines $H_1, \ldots, H_{p+1}$.

(2) Compute $W_i$ and $E_i$, $1 \leq i \leq p + 1$.

(3) Compute $NW_i$ and $NE_i$ using the recursions of Observation 8, $1 \leq i \leq p$.

(4) Compute $SW_i$ and $SE_i$ using the recursions of Observation 8, $1 \leq i \leq p$.

(5) Compute $E_i^*$ and $W_i^*$ for $1 \leq i \leq p$.

(6) (a) $W_i^* < E_i^*$ if and only if $R_i$ is in C-0. The collection of null-boxes in $R_i$ is the collection of boxes B in $R_i$ for which $W_i^* < B < E_i^*$. Place all null-boxes in $\beta^*$.

(b) $W_i^* = E_i^*$ if and only if $R_i$ is in C-1. The unique vertical connecting edge E in $R_i$ is the vertical edge E contained in the line $x = W_i^* = E_i^*$. Place E in $\beta^*$.

(c) $W_i^* > E_i^*$ if and only if $R_i$ is in C-2. The unique 2-box in $R_i$ is the box B such that $E_i^* < B < W_i^*$; place this box in $\beta^*$. All null-boxes, 2 boxes, and vertical connecting edges are now in $\beta^*$.

(7) For the horizontal line $H_i$, $1 \leq i \leq p + 1$;

(a) if $u_i = v_i$ then $H_i$ contains no horizontal connecting edge;

(b) if $u_i < v_i$ then a horizontal edge E contained in $H_i$ is a horizontal connecting edge if and only if it lies in $H_i$ between the vertical lines $x = u_i$ and $x = v_i$, and is not an edge of any null-box. Place each such E in $\beta^*$.

(8) $S^* = \beta^*$. Stop.

Figure 11 illustrates the use of steps 2 through 6 of the algorithm, together with the result of step 7.

Computational Efficiency Questions.

We now establish that the order of computational effort of any algorithm which constructs $S^*$ is at least n(log n), where the logarithm is to the base 2, and n is the number of existing facilities. We shall see that the order of computational effort of the row algorithm is n(log n). Hence the row algorithm is "optimal", in the sense that its order of computational effort is as small as that of any algorithm for constructing S*.

Given n distinct numbers, $a_1, \ldots, a_n$, it is known that (see pps. 159-170 of [3], or pps. 65-67 of [1]), the minimum number of comparisons needed to rank the numbers is of order n(log n). Now define existing facility locations $P_j = (a_j, a_j)$, $1 \leq j \leq n$, and consider the example problem of specifying S* for the existing facility locations. In order to state S*, let [1], ..., [n] be a permutation of 1, 2, ..., n such that

$a_{[j]} < a_{[j+1]}$, $1 \leq j \leq n - 1$, and define

$$S_j^* = \{(x, y): \ a_{[j]} \leq x, y \leq a_{[j+1]}\}, \ 1 \leq j \leq n - 1.$$

Each $S_j^*$ is a 2-box with SW and NE vertices of $P_{[j]}$ and $P_{[j+1]}$ respectively. It is easy to use the row algorithm to establish that

$$S^* = S_1^* \cup S_2^* \cup \ldots \cup S_{n-1}^*.$$

Any other correct algorithm to determine S* would also need to specify each $S_j^*$, and <u>each</u> $S_j^*$ <u>is completely determined by the numbers</u> $a_{[j]}$ and $a_{[j+1]}$, $a_{[j]} < a_{[j+1]}$, $1 \leq j \leq n - 1$. Hence in order to specify S* <u>any</u> correct algorithm must be able to sort the numbers $a_1, \ldots, a_n$, and so its order of computational effort is at least n(log n). (Note this example problem also establishes that any algorithm to construct S* is at least of order n when the Line Construction Procedure is considered not to be part of the algorithm, as S* is the union of n - 1 boxes.)

Now consider the row algorithm. Step (i) requires the ranking of n numbers, and so is of order n(log n). If $n_i$ is the number of existing facilities on $H_i$, finding $W_i$ and $E_i$ is equivalent to the problem of finding the maximum and minimum numbers in a collection of $n_i$ numbers, and it is known (see Pohl [4]) that this problem can be solved in $<(1.5)n_i - 2>$ comparisons at best (here $<y>$ denotes the smallest integer no less than $y$). Since $<(1.5)n_i - 2> \leq (1.5)(n_i - 1)$, an upper bound on the number of comparisons made in Step (2) is given by

$$\sum \{(1.5)(n_i - 1):\ 1 \leq i \leq p + 1\} = (1.5)n - (1.5)(p + 1).$$

Steps (3) through (6) each require 2p comparisons, for a total of 8 p comparisons. In the analysis (Observation 20) we establish that Step (7) can be accomplished in no more than $8(p + 1)$ comparisons. Thus Steps (2) through (7) require at most $(1.5)\,n - (1.5)(p + 1) + 8p + 8(p + 1) = (1.5)\,n + (6.5)(p + 1) + 8p \leq 16n$ comparisons. The minimum number of comparisons needed for Step (1) is $n <\log n> - 2^{<\log n>} + 1$, which is no greater than $n <\log n>$. Thus an upper bound on the number of comparisons made in Steps (1) through (7) is given by $n <\log n> + 16n$. Therefore, roughly speaking, Step (1) requires more effort than Steps (2) through (7) when $n <\log n> \geq 16n$, which is true when $n \geq 2^{16} = 65{,}536$. Hence the nonlinear effort of the first step does not predominate until n is "large", in which case the row algorithm is of order $n(\log n)$. In effect, setting up the problem (Step (1)) requires more effort than solving it when n is large!

Some idea of how the computational effort of the row algorithm compares with algorithms of order $n^2$ or $n^3$ can be obtained by examining Table 1. The comparisons in the table are conservative, in the sense that order $n^2$ and $n^3$ algorithms typically have a multiplicative constant greater than one.

## ANALYSIS

Diamond Intersection Lemma. Given diamonds $D(P_i, e_i)$, $i \in I \subset \{1, 2, \ldots, n\}$, with

$$D \equiv \cap \{D(P_i, e_i) : i \in I\} \neq \phi$$

if X is a point in D such that no point in D dominates X, than X is in S*.

Proof. Let Y be any point in the plane. If $Y \notin D$ there is an index $q \in I$ such that $Y \notin D(P_q, e_q)$, and thus $r(P_q, X) \leq e_q < r(P_q, Y)$, so Y does not dominate X. By hypothesis Y does not dominate X if $Y \in D$, so $X \in S^*$. (We shall refer to this lemma as DIL.)

Property 1. Every null-box is efficient.

Proof. Let B be any null-box and let $X \in B$. Since B is a null-box we may choose $P_1$, $P_2$, $P_3$ and $P_4$ in NE(B), NW(B), SW(B), and SE(B) respectively as illustrated in Figure 7 . Define $e_i \equiv r(X, P_i)$, and construct $D_i \equiv D(P_i, e_i)$, $1 \leq i \leq 4$. With $L_{13} \equiv D_1 \cap D_3$, $L_{24} \equiv D_2 \cap D_4$, we note $L_{13}$ is a SE-NW line segment, $L_{24}$ is a SW-NE line segment, and

$$D = L_{13} \cap L_{24} = \{X\}.$$

Thus no point in D dominates X. DIL thus implies $X \in S^*$. Since X is an arbitrary point in B, $B \subset S^*$.

Observation 9. Only a 1-box can be adjacent to a 2-box. Whenever a 1-box and 2-box are adjacent, the arrows in the boxes are parallel, and the arrow in the 1-box points towards the edge common to the 1-box and 2-box.

Proof. Let B be a 2-box, and let B' be a box adjacent to B. It is direct to verify that if B' is a null-box, an occupied direction of B' is contained in an unoccupied direction of B, which is impossible.

If B' is a 2-box, either an occupied direction of one box is contained in an unoccupied direction of the other, or else there is no $P_i$ on the line separating the boxes; both impossible situations. If B' is a 1-box, an occupied direction of B' is contained in an unoccupied direction of B except when the arrows in the two boxes are parallel, which completes the proof.

Observation 10. Let B be a 2-box, and let B' be a 1-box adjacent to B, and in the same row (column) of boxes as B. If B" is any other box in the same row (column) as B and B' such that B' lies between B and B", then B" is also a 1-box, and the arrows in B' and B" have the same label.

Proof. Let B be a 2-box. By Observation 9, we know that any box B' adjacent to B is a 1-box. We may assume that B, B', and B" are all in the same row. Also, due to symmetry, we may assume that the arrow labels of B are NE and SW, and that B' is east of B. Thus B" will be east of B', as illustrated in Figure 8. Now SW(B') $\subset$ SW(B") and SW(B') occupied implies SW(B") is occupied. NW(B) $\subset$ NW(B") and NW(B) occupied implies NW(B") is occupied. NE(B") $\subset$ NE(B) and NE(B) unoccupied implies NE(B") is unoccupied. If SE(B") is unoccupied, no $P_i$ would lie on the vertical line passing through the east edge of B", which is impossible. Thus SE(B") is occupied, and B" is a 1-box having an arrow with a NE label, the same label as the arrow in B'.

Property 2. Every 2-box is efficient.

Proof. Let B be any 2-box. Without loss of generality we may assume the arrows in B are NE and SW arrows, that B is the intersection of half-planes defined by lines as illustrated in Figure 9. If there is a 1-box B' which is adjacent and north of B, by Observations 9 and 10, the arrow in B' has a NE label. If there is an adjacent box B" to B such that B" is south of B, then the arrow in B" has a SW label. Since SW(B) is unoccupied, there must be some $P_i \in V \cap NW(B)$, or else there would be no $P_i \in V$. Likewise, there must be some $P_j \in V' \cap SE(B)$. Choose any $X \in B$, and define $e_i = r(X,P_i)$, $e_j = r(X,P_j)$, $D = D(P_j, e_i) \cap D(P_j, e_j)$. Since $X \in D$, by DIL $X \in S^*$ provided no point in D dominates X. We make the following observation: D is a SW-NE line segment, entirely contained in the column of boxes, say C, which contains B and B'. Now if $Y \in D \cap B$, $r(X, P_i) = r(Y, P_i)$ for all i, so Y does not dominate X, thus completing the proof for the case when $B' = \phi = B''$. If $Y \in D$, $Y \notin B$, Y is in a box in C which is a 1-box, and, hence, by Observations 9 and 10, there is a point Y' in $B \cap D$ which dominates Y. Since Y' dominates Y, and Y' does not dominate X, Y does not dominate X, so $X \in S^*$. Thus $B \subset S^*$.

<u>Property 3</u> Every connecting edge is efficient.

<u>Proof</u>. Let E be any connecting edge. Without loss of generality, we may assume E is an east edge of a 1-box B, and that B has a SW arrow.

Consider first the case where E is contained in the boundary of $\beta$. Then, as Figure 6a illustrates, since E lies on the east boundary of $\beta$, SW(B) is unoccupied, and there is a point $P_i \in H \cap V$. Also, since E lies on the east boundary of $\beta$ and NE(B) is occupied, there is a point $P_k \in V \cap NE(B)$. If $X \in E$, $e_i \equiv r(X, P_i)$, $e_k \equiv r(X, P_k)$, and $D \equiv D(P_i, e_i) \cap D(P_k, e_k)$, then $D = \{X\}$, and so by DIL, $X \in S^*$ and thus $E \subset S^*$.

Now suppose there is a 1-box B', adjacent to, and east of, B. The arrow in B' is either a SE arrow, as illustrated in Figure 6b, or a NE arrow, as illustrated in Figure 6c. In the former case we conclude there is some $P_i \in SE(B) \cap SW(B')$, some $P_j \in V' \cap NW(B)$, and some $P_k \in V'' \cap NE(B')$, as illustrated by Figure 6b. Hence if $X \in E$, defining $e_i = r(X, P_i)$,

$e_j = r(X, P_j)$, $e_k = r(X, P_k)$, $D = D(P_i, e_i) \cap D(P_j, e_j) \cap D(P_k, e_k)$, we conclude $D = \{X\}$ and thus, by DIL, X, and hence E, is efficient. In the latter case, NE(B') unoccupied and NE(B) occupied implies some $P_i \in NE(B) \cap V$. Likewise SW(B) unoccupied and SW(B') occupied implies some $P_k \in SW(B') \cap V$. Hence for any $X \in E$, with $e_i \equiv r(X, P_i)$, $e_k \equiv r(X, P_k)$, $D \equiv D(P_i, e_i) \cap D(P_k, e_k)$, we conclude $D = \{X\}$, invoke DIL, and conclude X, and thus E, is efficient.

<u>Observation 11</u>. If $X \in \bar{\beta}$ and lies at the intersection of four 1-boxes, then X lies on a connecting edge.

<u>Proof</u>. With reference to Figure 10, there exist 1-boxes, $B_1, \ldots, B_4$ such that $\{X\} = B_1 \cap B_2 \cap B_3 \cap B_4$. Define the edges $E_{12} = B_1 \cap B_2$, $E_{23} = B_2 \cap B_3$, $E_{34} = B_3 \cap B_4$, $E_{41} = B_4 \cap B_1$. We shall establish at least one of these edges is a connecting edge. Suppose none of the edges is a connecting edge. Then for each edge $E_{ij}$ at least one arrow in $B_i$ or $B_j$ does not point towards $E_{ij}$. Without loss of generality, suppose the arrow $\bar{\alpha}_1$ in $B_1$ does not point towards $E_{14}$, so that $\bar{\alpha}_1$ is not a NE or a NW arrow. $\bar{\alpha}_1$ cannot be a SW arrow, for then X would not be in $\bar{\beta}$. Thus $\bar{\alpha}_1$ is a SE arrow, and points toward $E_{12}$. The arrow $\bar{\alpha}_2$ in $B_2$ cannot be a SE arrow as then X would not be in $\bar{\beta}$. Since $\bar{\alpha}_1$ points towards $E_{12}$, $\bar{\alpha}_2$ cannot be a SW or NW arrow, and thus $\bar{\alpha}_2$ is a NE arrow. Similarly, we conclude $\alpha_3$ is a NW arrow, and $\bar{\alpha}_4$ is a SW arrow. But now we have the situation illustrated in Figure 10, where an occupied direction of each box is contained in an unoccupied direction of an adjacent box. Such a situation is impossible, and so X lies on at least one connecting edge.

<u>Property 3</u>. $\bar{\beta}$ is contained in $\beta^*$.

<u>Proof</u>. Let $X \in \bar{\beta}$. Since every box in $\beta$ is a null-box, 1-box, or 2-box,

X must lie in one such box. If the box is a null-box or 2-box, certainly $X \in \beta^*$, so it remains to consider the case where X is in a 1-box, but X is in no null-box or 2-box.

Since X is in a 1-box, say $B_1$, and in $\bar{\beta}$, X must lie on some leading edge of $B_1$, say E, of positive length. If E is contained in the boundary of $\beta$, since the arrow in $B_1$ points towards E, E is a connecting edge. Thus it remains to consider the case where E is not contained in the boundary of $\beta$. In this case there is some box, say $B_2$, such that $E = B_1 \cap B_2$. Thus X is in $B_2$, and so $B_2$ must be a 1-box. If X is not an endpoint of E, since $X \in \bar{\beta}$ the arrows in $B_1$ and $B_2$ must point towards E, and so E is a connecting edge. Thus suppose X is at an endpoint of E. Since $X \in B_1 \cap B_2$ and X is not on the boundary of $\beta$, there exist 1-boxes $B_3$ and $B_4$ such that $\{X\} = B_1 \cap B_2 \cap B_3 \cap B_4$ (as illustrated in Figure 10 ). Thus, by Observation 11, X lies on a connecting edge.

<u>Theorem 1</u>. $\bar{\beta}$, $\beta^*$, and $S^*$ are identical and nonempty.

<u>Proof</u>. By Observation 6 , every point not in $\bar{\beta}$ is dominated, and thus $S^* \subset \bar{\beta}$. By Properties 1, 2, and 3, $\beta^* \subset S^*$, and so $\beta^* \subset S^* \subset \bar{\beta}$. Since Property 4 gives $\bar{\beta} \subset \beta^*$, we conclude $\beta^* = S^* = \bar{\beta}$. As every $P_i$ is in $S^*$, $S^* \neq \phi$, so $\beta^* = \bar{\beta} = S^* \neq \phi$.

We now give the analysis needed to justify the row algorithm.

Observation 12. A box B in $R_i$ is a null-box if and only if $W_i^* < B < E_i^*$.

Proof. By Observation 7, the four directions of a null-box B are occupied if and only if $NW_i < B$, $SW_i < B$, $B < NE_i$, $B < SE_i$, which is equivalent to $W_i^* < B < E_i^*$.

Observation 13. A box B in $R_i$ is a 2-box if and only if either $NE_i < B < SE_i$ or $NW_i < B < SW_i$.

Proof. By Observation 3, a box B in $R_i$ is a 2-box if and only if B is either a NW-SE or a NE-SW 2-box. B in $R_i$ is a NE-SW 2-box if and only if NE(B) and SW(B) are unoccupied and NW(B) and SE(B) are occupied. Thus Observation 7 implies B is a NE-SW 2-box if and only if $NW_i < B$, $NE_i < B$, $B < SW_i$, $B < SE_i$. By definition, $NW_i \leq NE_i$ and $SW_i \leq SE_i$, so B in $R_i$ is a NE-SW 2-box if and only if $NE_i < B < SW_i$. Similarly, B in $R_i$ is a NW-SE 2-box if and only if $SE_i < B < NW_i$.

Observation 14. Given a box B in $R_i$,

$$NE_i < B < SW_i \qquad \text{(i)}$$

or

$$SE_i < B < NW_i \qquad \text{(ii)}$$

if and only if

$$E_i^* < B < W_i^*. \qquad \text{(iii)}$$

Proof. By definition, $E_i^* \leq SE_i$, $E_i^* \leq NE_i$, $SW_i \leq W_i^*$, and $NW_i \leq W_i^*$, so if (i) or (ii) is true then (iii) is true.

Suppose (iii) is true. Either $E_i^* = SE_i$ or $E_i^* = NE_i$. When $E_i^* = SE_i$, (iii) and $SW_i \leq SE_i$ give

$$SW_i \leq SE_i = E_i^* < B < \max(SW_i, NW_i). \qquad \text{(iv)}$$

Since (iv) implies $B < NW_i$, we conclude (iii) implies (ii). When $E_i^* = NE_i$, (iii) and $NW_i \leq NE_i$ give

$$NW_i \leq NE_i = E_i^* < B < \max(SW_i, NW_i). \tag{v}$$

Since (v) implies $B < SW_i$, we conclude (iii) implies (ii).

We summarize Observations 9, 10, 13 and 14 in

Observation 15. A box B in $R_i$ is a 2-box if and only if $E_i^* < B < W_i^*$. $R_i$ contains exactly one 2-box if and only if $E_i^* < W_i^*$.

Next we state a readily established result needed in identifying vertical connecting edges.

Observation 16. (a) If B is a westmost NE(SE) 1-box in $R_i$, then every box east of B in $R_i$ is a NE(SE) 1-box, while if B' is any box in $R_i$ which has a label and is west of B, then the label of B' is either NW or SW. (b) If B is an eastmost NW(SW) 1-box in $R_i$, then every box west of B in $R_i$ is a NW(SW) 1-box, while if B' is any box in $R_i$ which has a label and is east of B, then the label of B' is either NE or SE. (c) The labels of 1-boxes in any row are of at most two different types.

Observation 17. Given a vertical edge E in $R_i$ lying on the vertical line $x = V$, E is a vertical connecting edge if and only if $W_i^* = V = E_i^*$.

Proof. Due to symmetry it is enough to consider the cases $W_i^* = SW_i$, $E_i^* = NE_i$, and $E_i^* = SE_i$, as illustrated in Figure 6. Also we may assume there is a box B in $R_i$ such that E is the east edge of B. If E lies on the line $x = V = SW_i = W_i^*$, B has only one arrow, a SW arrow, which points towards E. Thus if E is contained in the east boundary of $\beta$ then E is a vertical connecting edge. Otherwise there is a box B' in $R_i$, east of B, such that $E = B \cap B'$. Since E lies on the line $x = V$ and $V = SE_i$ or $V = NE_i$, there is exactly one arrow in B', a SE or NE arrow, which points towards E. Thus $E_i^* = V = W_i^*$ implies E is a vertical connecting edge.

Conversely, let E be a vertical connecting edge in $R_i$ contained in the line $x = V$. If $E_i^* \neq W_i^*$, Observations 12 and 15 would imply $R_i$ is in C − 0 or C − 2, in which case Observation 16, or Observations 9 and 10, would imply an arrow of a box of which E is an edge does not point towards E, contradicting the fact that E is a vertical connecting edge. Thus $W_i^* = E_i^*$.

It remains to show $W_i^* = V = E_i^*$. Let $x = V'$ be a line such that $W_i^* = V' = E_i^*$ and suppose $V \neq V'$. Since $W_i^* = V'$, every box west of V' will have either a NW or SW label, while since $V' = E_i^*$ every box east of V' will have either a NE or SE label. Denote by E' the vertical edge in $R_i$ contained in V'. From the first part of the proof we know E' is a vertical connecting edge. Without loss of generality we may assume V' is west of V. Thus there exists a 1-box B' of $R_i$ such that E' is a west edge of B', and there exists a 1-box B of $R_i$ such that E is an east edge of B. Since E and E' are vertical connecting edges, there is a NW or SW arrow in B pointing towards E, and a NE or SE arrow in B' pointing towards E'. Further, NE(B) and SE(B) are occupied while either SE(B') or NE(B') is unoccupied. But NE(B) and SE(B) are contained in SE(B') and NE(B') respectively. Thus an occupied direction of B is contained in an unoccupied direction of B' if $V \neq V'$, and we have a contradiction. Thus $W_i^* = V = E_i^*$.

Given any horizontal edge E, whenever E is east of the vertical line $x = V$ and west of the vertical line $x = V'$ we write $V < E < V'$. Given any edge E contained in $H_i$, it is easy to verify that E is uncolored if and only if $u_i < E < v_i$. Hence the following result can be established:

Observation 18. Given an edge E contained in a horizontal line $H_i$, E is a horizontal connecting edge if and only if $u_i < E < v_i$ and E is not an edge of any null-box.

We now state

Theorem 2. The set constructed by the row algorithm is the efficient set.

Proof. A direct consequence of Observations 12, 15, 17, 18, and Theorem 1.

It now only remains to consider the computational effort involved in Step (7) of the row algorithm. Due to Observation 18, if $u_i = v_i$ there is no connecting edge in $H_i$. If $u_i < v_i$ and we remove from $H_i$ the interior points of all edges which are null-box edges, then the remaining edges (if any) are the horizontal connecting edges in $H_i$. We consider the case where there are null-boxes in both $R_{i-1}$ and $R_i$, as this case requires the most effort. Let $I \equiv [u_i, v_i]$, $J' \equiv [W^*_{i-1}, E^*_{i-1}]$, $J'' \equiv [W^*_i, E^*_i]$. I is the x projection of the uncolored portion of $H_i$, while, by Observation 12, J' and J" are the x-projections of the null-boxes in $R_{i-1}$ and $R_i$ respectively. Because the null-boxes whose horizontal edges are in $H_i$ have these edges uncolored, we have $J' \subset I$, $J'' \subset I$.

The most direct way to determine the computational effort for this case is to state a simple algorithm which identifies all the horizontal connecting edges in $H_i$. We consider the algorithm self-evident, as it simply determines all the points in I which are not interior to $J' \cup J''$.

Observation 19. When $R_{i-1}$ and $R_i$ each contain null-boxes, the horizontal connecting edges in $H_i$ may be determined as follows.

(a) check to see which of the intervals J' and J" has a leftmost endpoint; denote this interval by I' and denote the other interval by I". Let $I' = [a', b']$, $I'' = [a'', b'']$.

(b) Check to see if I' abuts the left endpoint of I; if it does not, the edges in $H_i$ whose x projections lie in $[u_i, a']$ are horizontal connecting edges.

(c) Check to see if I' and I" intersect; if they do not, the edges in $H_i$ whose x projections lie in $[b', a'']$ are horizontal connecting edges.

(d) Check to see which of the two intervals I' and I" has a rightmost right endpoint, and denote this interval by $I''' = [a''', b''']$.

(e) Check to see if I"' abuts the right endpoint of I; if it does not, the edges in $H_i$ whose x projections lie in $[b''', v_i]$ are horizontal connecting edges.

In steps (a) through (e) respectively we note that the following terms are compared: $W^*_{i-1}$ and $W^*_{i}$; $u_i$ and a'; b' and a"; b' and b"; b"' and $v_i$. Hence, given all the necessary data, we can find all the horizontal connecting edges by making at most 5(p + 1) comparisons. To obtain the data, with reference to Step (7), we must compute the $u_i$ and $v_i$ (2 p comparisons) and then compare them (p + 1 comparisons). Thus Step (7) requires at most

$5(p + 1) + 2p + (p + 1) \leq 8(p + 1)$ comparisons. As $p + 1 \leq n$, we have

Observation 20. Step (7) of the row algorithm requires at most $8(p + 1) \leq 8n$ comparisons.

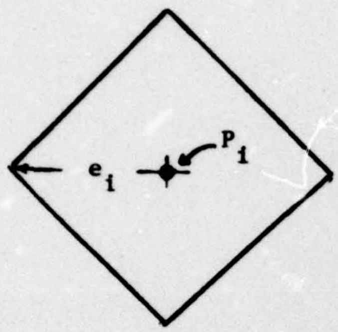



Figure 1: $D(P_i, e_i)$

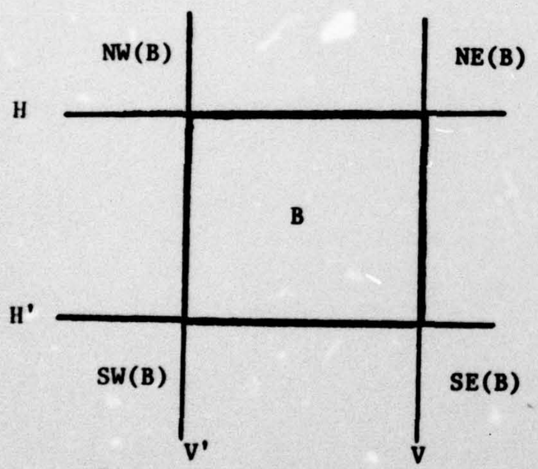



Figure 3: Directions of B

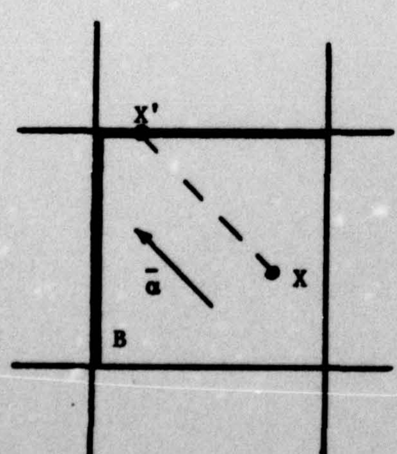



Figure 4: Leading Edges of SE 1-Box

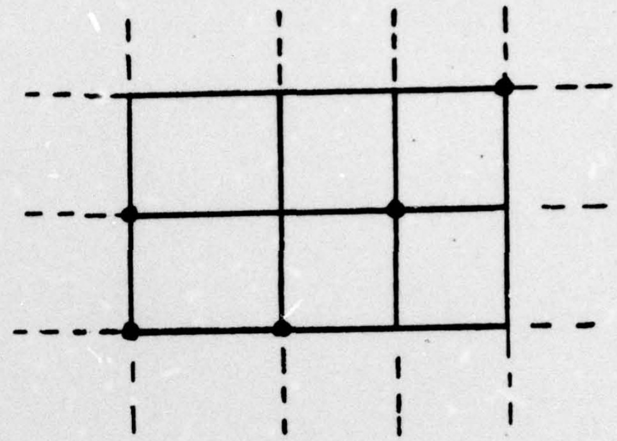


Figure 2a: Line Construction

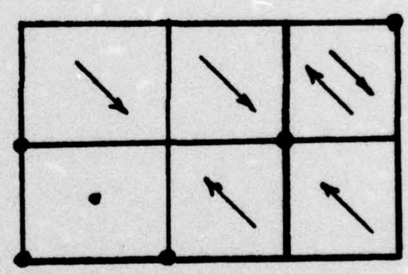


Figure 2b: Arrow Drawing

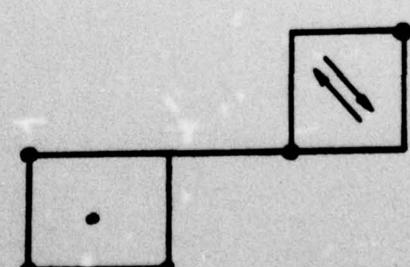


Figure 2c: $\bar{\beta} = S^* = \beta^*$

Figure 2: Use of Arrow Algorithm

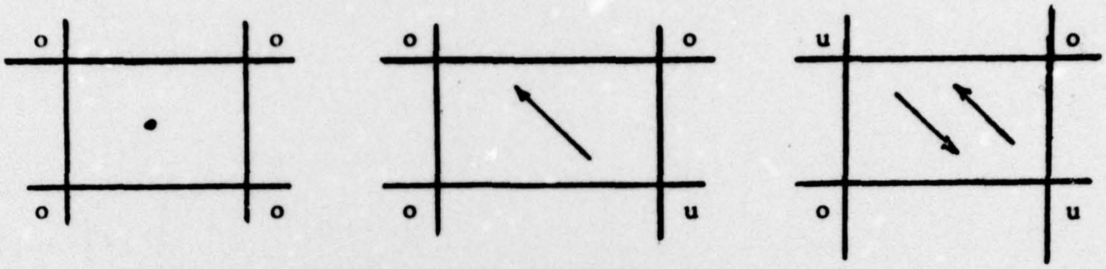



Figure 5: Null-Box, 1-Box, and 2-Box Examples
(o: occupied, u: unoccupied)

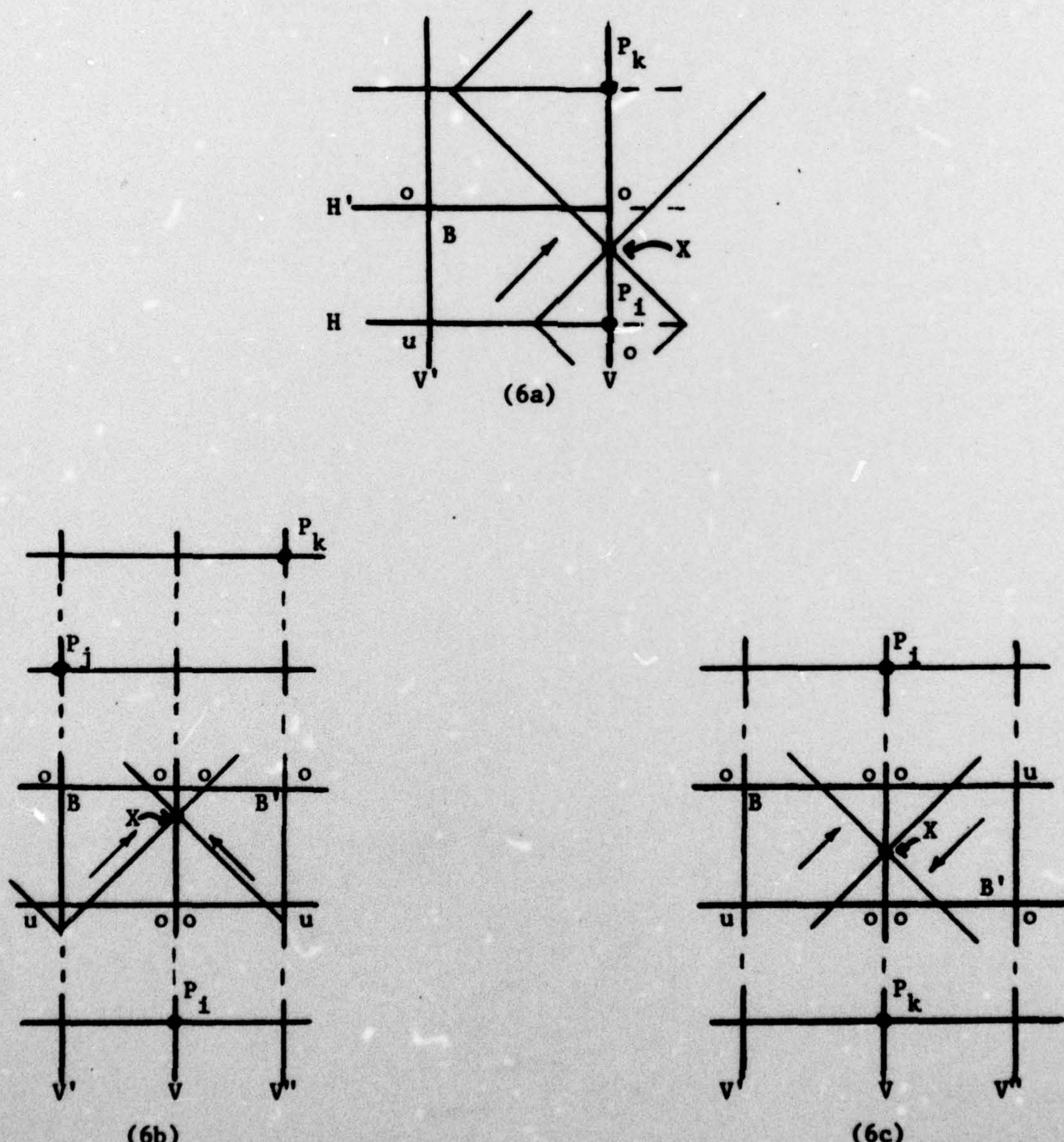



Figure 6: Efficiency of Leading Edges

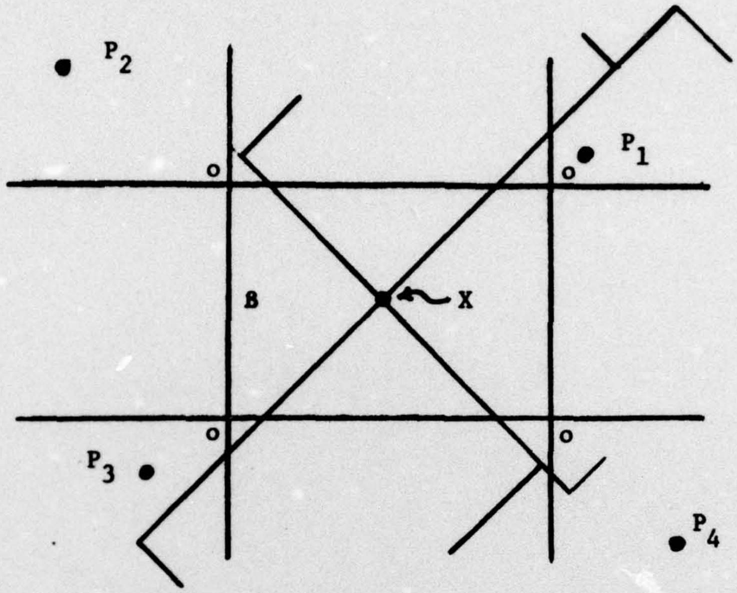



Figure 7: Efficiency of Null-Boxes

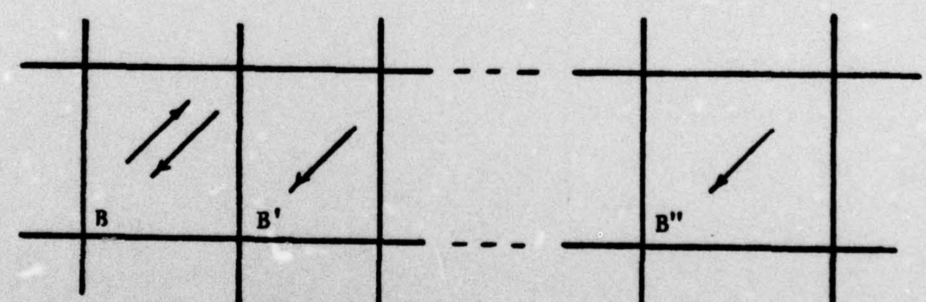



Figure 8: At Most One 2-Box Per Row

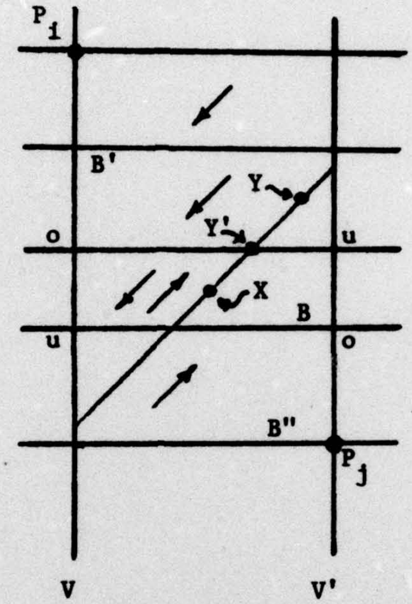



Figure 9: Efficiency of 2-Boxes

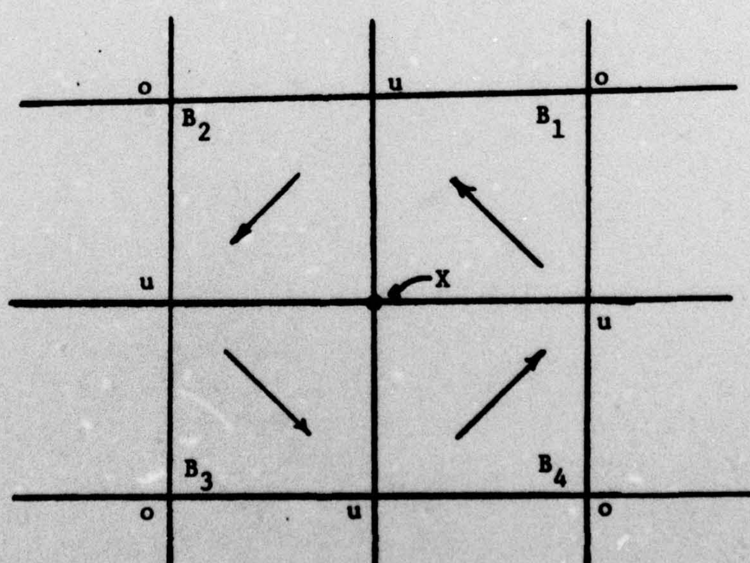
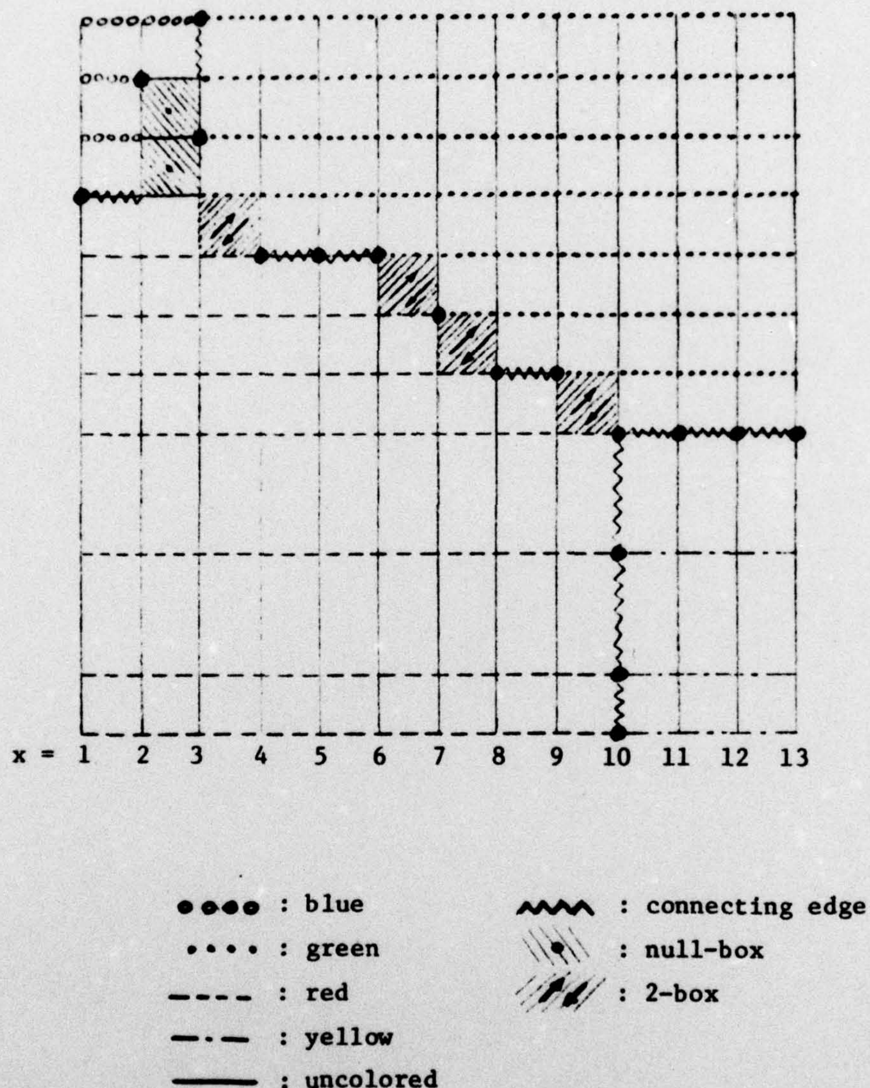



Figure 10: $\{X\} = B_1 \cap B_2 \cap B_3 \cap B_4$

| $W_i^*$ | $NW_i$ | $SW_i$ |
|---|---|---|
| 3 | 3 | 1 |
| 2 | 2 | 1 |
| 2 | 2 | 1 |
| 4 | 1 | 4 |
| 7 | 1 | 7 |
| 8 | 1 | 8 |
| 10 | 1 | 10 |
| 10 | 1 | 10 |
| 10 | 1 | 10 |
| 10 | 1 | 10 |




| $SE_i$ | $NE_i$ | $E_i^*$ |
|---|---|---|
| 13 | 3 | 3 |
| 13 | 3 | 3 |
| 13 | 3 | 3 |
| 13 | 3 | 3 |
| 13 | 6 | 6 |
| 13 | 7 | 7 |
| 13 | 9 | 9 |
| 10 | 13 | 10 |
| 10 | 13 | 10 |
| 10 | 13 | 10 |

**Figure 11: Row Algorithm Example**

| $n$ | $n^2$ | $n^3$ | $r(n)$ |
|---|---|---|---|
| 2 | 4 | 8 | 33 |
| 4 | 16 | 64 | 69 |
| 8 | 64 | 512 | 145 |
| 16 | 256 | 4,096 | 305 |
| 32 | 1,024 | 32,768 | 641 |
| 64 | 4,096 | 262,144 | 1,345 |
| 128 | 16,384 | 2,097,152 | 2,817 |
| 256 | 65,536 | 16,777,216 | 5,889 |

Table 1: Computational Effort Comparisons for $n^2$, $n^3$, and Row Algorithm,

$r(n) \equiv n < \log n > - 2^{< \log n >} + 1 + 16 n$

## REFERENCES


[1] Aho, Alfred V, Hopcroft, John E., and Ullman, Jeffrey D., The Design and Analysis of Computer Algorithms, Addison-Wesley Publishing Co., Reading Mass., 65-67 (1974).

[2] Geoffrion, Arthur M., "Proper Efficiency and the Theory of Vector Maximization," Journal of Mathematical Analysis and Applications, 22, 3 (June 1968).

[3] Knuth, Donald E., The Art of Computer Programming, Vol. 3, Sorting and Searching, Addison-Wesley Publishing Co., Reading, Mass., 159-170 (1973).

[4] Pohl, I., "A Sorting Problem and Its Complexity," Communications of the Association of Computing Machinery, 15, 6 (December 1972).

[5] Shamos, Michael Ian, and Hoey, Dan, "Closest-Point Problems," Proc. 16th Annual Symposium on Foundations of Computer Science (IEEE) held at Berkeley, CA, Oct. 13-15, 151-162 (1975).

[6] Shamos, Michael Ian, "Geometric Complexity," ACM Symposium on Theory of Computing, 224-233 (1976).

[7] Rockafellar, R. T., Convex Analysis, Princeton University Press, Princeton, New Jersey (1970).

[8] Wendell, Richard E., Hurter, Arthur P., and Lowe, Timothy J. "Efficient Points in Location Problems," to appear in AIIE Transactions.